第3章

数据准备

视频讲解

【学习目标】

- 理解数据准备工作的步骤；
- 了解数据的不同类型；
- 理解数据处理的主要方法。

3.1 数据准备的基本知识

数据准备阶段包含从原始数据到形成最终数据集(将要被输入模型工具的数据)的所有操作。数据准备任务可能要进行多次,没有规定的固定顺序。任务包括：表、记录和特征的选择；数据的清洗和转换等。

《纽约时报》曾有一篇文章报道,数据科学家在挖掘出有价值的“金块”之前要花费50%～80%的时间在很多诸如收集数据和准备不规则的数据的普通任务上。处理混乱的数据是数据科学家工作流中典型的比较耗费时间的工作,如图3-1所示。

数据很少是“干净的”,经常有质量问题。例如数据的不唯一性、格式上不统一、非法值、特征依赖、缺失值、拼写错误、错位值等,如图3-2所示。

数据为什么会有质量问题？原因主要有以下几方面。

(1) 数据的不完全性：数据缺少特征或者包含缺失值；

(2) 数据噪声：数据包含错误的记录或者异常值；

(3) 数据的不一致性：数据包含冲突的记录或者存在差异。

影响数据质量的问题总结如下。

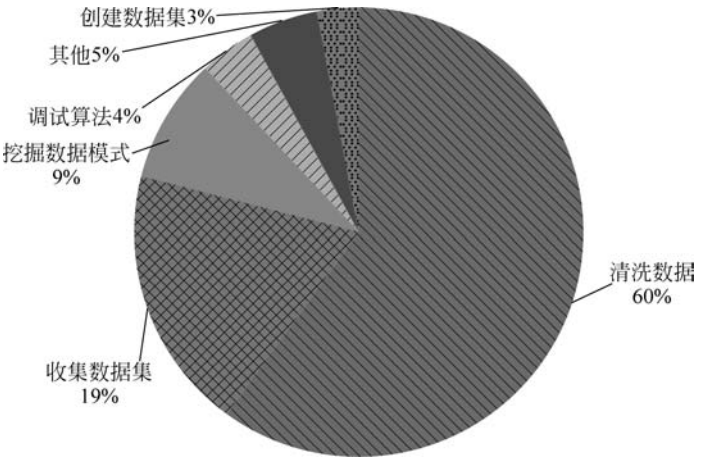

图 3-1　数据处理的时间分配

#	id	Name	Birthday	Gender	isTeacher?	#Students	Country	City
1	111	John	31/12/1990	M	0	0	Ireland	Dublin
2	222	Mery	15/10/1978	F	1	15	Iceland	
3	333	Alice	19/04/2000	M	0	0	Spain	Madrid
4	444	Mark	01/11/197	A	0	0	France	Paris
5	555	Alex	15/03/2000	M	1	23	Germany	Berlin
6	555	Peter	1983-12-01	M	1	10	Italy	Rome
7	777	Calvin	05/05/1995	F	0	0	Italy	Italy
8	888	Proxane	13/08/1948	F	0	0	Portugual	Lisbon
9	999	Anne	05/09/1992	F	0	5	Switzerlan	Genevo
10	101010	Paul	14/11/1992	M	1	26	Vidad	Rome

缺失值

不唯一性　格式不统一　非法值　特征依赖　拼写错误　错位值

图 3-2　有质量问题的数据

1. 非法值

一些数据集包含一些明显的值，例如性别只能是男或女，上面的例子很容易发现错误。

2. 格式

格式是最常见的问题。相同的数据可能获得不同格式的值，例如名字写成“姓名，xxx”或者“xxx 姓名”。

3. 特征依赖

一个特征的值依赖于另一个特征。例如，如果我们有一些学校数据，学生的数量与这个人是否是教师有关。如果某个人不是教师，他不可能有一些学生。

4. 唯一性

很可能发现只允许唯一值的数据重复。例如我们的两个产品不能有两个身份 ID。

5. 缺失值

数据集的一些特征的值可能是空白或者缺失的。

6. 拼写错误

拼写错误主要指英文等字符串拼写的错误。

7. 错位值

一个特征的值包含了另一个特征。

数据分析首先要保证导入的数据是“干净的”，才能得到有价值的信息。所以数据准备工作是非常重要的。

数据准备主要包括：数据归一化、数据离散化、文本清洗、数据清洗及数据降维，如图 3-3 所示。

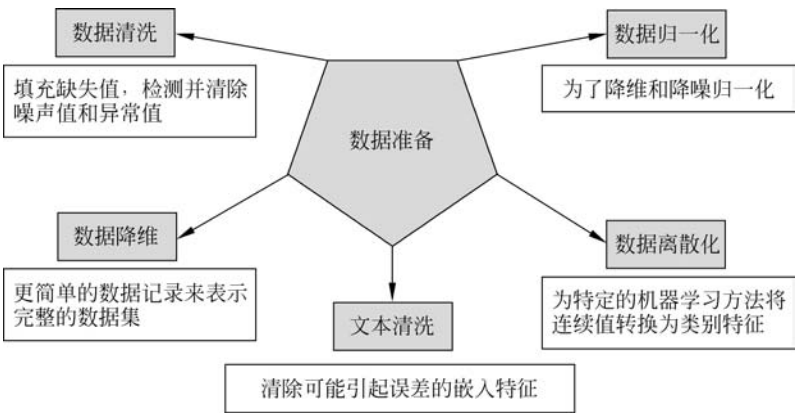

图 3-3　数据准备的主要工作

下面将对每个工作进行详细的介绍。

3.2　数据类型及处理方式

数据(Data)，是事实或观察的结果，是对客观事物的逻辑归纳，是用于表示客观事物的未经加工的原始素材。数据可以是连续的值，例如声音、图像，称为模拟数据；也可以是离散的，如符号、文字，称为数字数据。

数据分析的基础是数据分类，不同类型的数据有不同的处理方式。数据分类就是把具有某种共同属性或特征的数据归并在一起，通过其类别的属性或特征来对数据进行区别。换句话说，就是相同内容、相同性质的信息以及要求统一管理的信息集合在一起，而把相异的和需要分别管理的信息区分开来，然后确定各个集合之间的关系，形成一个有条理的分类系统。

3.2.1　统计计量角度的数据类型

从统计计量层面的角度看，可以将数据分为定类数据、定序数据、定距数据、定比数据，如表 3-1 所示。

表 3-1　四类数据的特征及举例

数据类型	特　　征	运算功能	举　　例
定类数据	分类	频数、频率、众数、异众比率	产业分类
定序数据	分类、排序	计数、排序、中位数，等级相关系数，非参数分析	企业等级
定距数据	分类、排序、有基本测量单位	回归分析、各种统计量、参数估计、假设检验等	产品质量差异
定比数据	分类、排序、有基本测量单位、有绝对零点	回归分析、各种统计量、参数估计、假设检验等	商品销售额

1. 定类数据

定类就是给数据定义一个类别。这种数据类型将所研究的对象分类，也即只能决定研究对象是同类或不同类。例如把性别分成男女两类；把动物分成哺乳类和爬行类等。

定类变量遵循以下两个原则。

(1) 类与类之间互斥，不存在既是男又是女的性别。

(2) 每个对象都必须有类别，就像动物按照域(Domain)、界(Kingdom)、门(Phylum)、纲(Class)、目(Order)、科(Family)、属(Genus)、种(Species)加以分类。

2. 定序数据

定序数据是将同一个类别下的对象分一个次序，即变量的值能把研究对象排列高低或大小，具有＞与＜的数学特质。它是比定类变量层次更高的变量，因此也具有定类变量的特质，即区分类别(＝、≠)。例如文化程度可以分为大学、高中、初中、小学、文盲；工厂规模可以分为大、中、小；年龄可以分为老、中、青。这些变量的值，既可以区分异同，也可以区别研究对象的高低或大小。

各个定序变量的值之间没有确切的间隔距离。例如大学究竟比高中高出多少，大学与高中之间的距离和初中与小学之间的距离是否相等，通常是没有确切的尺度可以测量的。定序变量在各个案上所取的变量值只具有大于或小于的性质，只能排列出它们的顺序，而不能反映出大于或小于的数量或距离。

3. 定距数据

定距数据是区别同一类别下各个案中等级次序及其距离的变量。它除了包括定序变量的特性外，还能确切测量同一类别各个案高低、大小次序之间的距离，因而具有加与减的数学特质。但是，定距变量没有一个真正的零点。例如，摄氏温度(℃)这一定距变量说明，40℃比 30℃高 10℃，30℃比 20℃又高 10℃，它们之间高出的距离相等，而 0℃并不是

没有温度。又例如调查数个地区的劳动人口增长比率时，发现甲、乙，丙、丁、戊五个地区的比率分别是 2%、10%、35%、20%、10%。这不能说明甲和戊增长的人数是相同的，也不能说明丁增长的人数是戊的两倍。但是我们可以从数据中看出甲和戊的增长速度一样，丁的增长速度是戊的两倍。

定距变量各类别之间的距离，只能加减而不能乘除或以倍数的形式来说明它们之间的关系。

4. 定比数据

定比数据是区别同一类别个案中等级次序及其距离的变量。定比变量除了具有定距变量的特性外，还具有一个真正的零点，因而它具有乘与除(×、÷)的数学特质。例如年龄和收入这两个变量，既是定距变量，同时又是定比变量，因为其零点是绝对的，可以作乘除的运算。如 A 月收入是 60 元，而 B 是 30 元，我们可以算出前者是后者的两倍。而智力商数这个变量是定距变量，但不是定比变量，因为其 0 分只具有相对的意义，不是绝对的或固定的，不能说某人的智商是 0 分就是没有智力；同时，由于其零点是不固定的，即使 A 是 140 分而 B 是 70 分，我们也不能说前者的智力是后者的两倍，只能说两者相差 70 分。因为 0 值是不固定的，如果将其提高 20 分，则 A 的智商变为 120 分而 B 变成 50 分，两者的相差仍是 70 分，但 A 却是 B 的 2.4 倍，而不是原先的两倍了。摄氏温度这一变量也是如此。

定比变量是最高测量层次的变量。

不同类型的数据有不同的运算功能，如表 3-1 所示，不同数据的统计处理方式也不同。

(1) 定类数据可以进行的统计处理方式包括：频数、频率、众数、异众比率等的计算。

(2) 定序数据可以进行的统计处理方式包括：中位数、等级相关系数、非参数分析等的计算。

(3) 定距数据、定比数据可以进行的统计处理方式有：回归分析、各种统计量、参数估计、假设检验等的计算。

不同统计处理方式对数据要求不同，一般来说，等级高的数据兼有等级低的数据功能，等级低的数据没有等级高的数据功能。

在对数据进行可视化展示时，不同类型数据也应采用合适的图形。

一般对于定类、定序数据，适合使用条形图和饼形图。其中用宽度相同的条形的长短来表示数据变动的图形为条形图；用圆形及圆内扇形面积表示数值大小的图形为饼形图。

而定距、定比数据适合采用直方图、折线图和茎叶图。其中用矩形的宽度和高度来表示频数的图形为直方图；在直方图基础上，把直方图顶部的中点用直线连接起来形成的折线图为折线图；对于未分组的原始数据，用茎叶图来显示分布特征，统称为茎叶图。

3.2.2 计算机角度的数据类型

在计算机中，用变量来存储数据，它们有名字和数据类型。变量的数据类型决定了如

何将代表这些值的位存储到计算机的内存中。在声明变量时也可指定它的数据类型。所有变量都具有数据类型，以决定能够存储哪种数据。

数据类型包括原始类型、多元组、记录单元、代数数据类型、抽象数据类型、参考类型以及函数类型。

计算机中，以位(0 或 1)表示数据。数据的最小的寻址单位称为字节(通常是八位)。机器码指令处理的单位，称作字长。大部分对字长的指令解译，主要以二进制为主，如一个 32 位的字长，可以表示从 0 至 2 的 32 次方减 1 的无符号整数值，或者表示从负的 2 的 32 次方至 2 的 32 次方减 1 的有符号整数值。存在特殊的算术指令，对字长中的位使用不同的解释，以此作为浮点数。

数据类型的出现是为了把数据分成所需内存大小不同的数据，在编程需要使用大数据时才需要申请大内存，这时就可以充分利用内存。除变量外，数据类型也用于其他场合。在给属性赋值时，这个值就有数据类型；函数的参数也有数据类型。数据类型分为如下几个类型。

1. 数字型

一般的编程语言或软件支持以下几种数字类型：Integer(整型)、Long(长整型)、Single(单精度浮点型)、Double(双精度浮点型)和 Currency(货币型)。此处列举的是 Visual Basic 里的类型，其他语言类似，只是名称有所差别。

如果知道变量总是存放整数(如 12)而不是带小数点的数字(如 3.57)，就应当将它声明为 Integer 类型或 Long 类型。整数的运算速度较快，而且比其他数据类型占据的内存要少。在 For...Next 循环内作为计数器变量使用时，整数类型尤为有用。

如果变量包含小数，则可将它们声明为 Single、Double 或 Currency 变量。Currency 数据类型支持小数点右面 4 位和小数点左面 15 位，它是一个精确的定点数据类型，适用于货币计算。浮点(Single 和 Double)数比 Currency 的有效范围大得多，但有可能产生小的进位误差。

浮点数值可表示为 mmmEeee 或 mmmDeee，其中 mmm 是假分数，而 eee 是指数(以 10 为底的幂)。Single 数据类型的最大正数值为 3.402823E＋38，或 3.4×10^{38}；Double 数据类型的最大正数值是 1.79769313486232D＋308 或 1.8×10^{308}。用 D 将数值文字中的假数部分和指数部分隔开，就会导致将该值作为 Double 数据类型来处理。同样，用这种方式使用 E，也会导致将该值作为 Single 数据类型来处理。

2. 字节型

如果变量包含二进制数，则将其声明为 Byte 数据类型的数组。在转换格式期间用 Byte 变量存储二进制数据就可保留数据。当 String 变量在 ANSI 和 Unicode 格式间进行转换时，变量中的任何二进制数据都会遭到破坏。除一元减法外，所有可对整数进行操作的运算符均可操作 Byte 数据类型。因为 Byte 是从 0～255 的无符号类型，所以不能表示负数。所有数值变量都可相互赋值。

3. 文本型

如果变量总是包含字符串而从不包含数值，就可将其声明为 String 类型（文本型）。按照默认规定，String 变量或参数是一个可变长度的字符串，随着对字符串赋予新数据，它的长度可增可减。也可以声明字符串具有固定长度。

4. 逻辑型

若变量的值只是 true/false、yes/no、on/off 信息，则可将其声明为 Boolean 类型（逻辑/布尔型）。Boolean 的默认值为 False。

5. 日期型

日期型包含 Date（日期型）和 Time（时间型）两种数据类型，一般的 Date 特性适用于这两种类型。当其他数值数据类型转换为 Date 时，小数点左边的值表示 Date 信息，小数点右边的值则代表 Time。午夜为 0，正午为 0.5。负数表示公元 1899 年 12 月 31 日之前的 Date。

6. 对象型

Object 变量（对象型）作为 32 位（4 字节）地址来存储，该地址可引用应用程序中或某些其他应用程序中的对象。可以随后（用 Set 语句）指定一个被声明为 Object 的变量去引用应用程序所识别的任何实际对象。

7. 变体型

Variant 变量（变体型）能够存储所有系统定义类型的数据。如果对 Variant 变量进行数学运算或函数运算，则 Variant 必包含某个数。

表 3-2 显示了计算机所支持的数据类型，以及存储空间的大小与范围。

表 3-2 计算机支持的数据类型、存储空间的大小与范围

数 据 类 型	储存空间大小/字节	范　　围
Byte（字节型）	1	0～255
Boolean（布尔型/逻辑型）	2	true 或 false
Integer（整型）	2	－32 768～32 767
Long（长整型）	4	－2 147 483 648～2 147 483 647
Single（单精度浮点型）	4	负数范围： －3.402 823E38～－1.401 298E－45 正数范围： 1.401 298E－45～3.402 823E38

续表

数 据 类 型	储存空间大小/字节	范　　围
Double(双精度浮点型)	8	负数范围： －1.797 693 134 862 32E308～－4.940 656 458 412 47E-324 正数范围： 4.940 656 458 412 47E-324～1.797 693 134 862 32E308
Currency(货币型)	8	－922 337 203 685 477.5808～922 337 203 685 477.5807
Decimal(十进制型)	14	没有小数点时： ＋/－79 228 162 514 264 337 593 543 950 335 有小数点时： ＋/－7.922 816 251 426 433 759 354 395 033 5 最小的非零值： ＋/－0.000 000 000 000 000 000 000 000 000 1
Date、Time(时间日期型)	8	100 年 1 月 1 日～9999 年 12 月 31 日
Object(对象型)	4	任何 Object 引用
String(文本型)(变长)	10	长度从 0～20 亿
String(文本型)(定长)	10	长度从 1～65 400
Variant(变体型)(数字)	16	任何数字值，最大可达 Double 的范围
Variant(变体型)(字符)	22	与字符串长度，变长 String 有相同的范围

3.2.3　数据处理方式

一般可将数据分为离散型数据、连续型数据、时间数据、空间数据，其对应的数据处理方式也不相同[①]。

1. 离散型数据

在进行机器学习或深度学习的建模中，总会碰到离散型数据，例如，性别：男，女；学历：高中、大学、硕士、博士。一般来说，对离散型的数据有以下几种处理方式。

1) One-Hot Encoding：One-Hot 编码

又称为一位有效编码，主要采用位状态寄存器来对各状态进行编码，每个状态都有它独立的寄存器位，并且在任意时候只有一位有效。这种方式下，若特征种类很多，则呈现出高稀疏化特征。例如，有如下三个特征属性。

(1) 性别：["male","female"]。

(2) 地区：["Europe","US","Asia"]。

(3) 浏览器：["Firefox","Chrome","Safari","Internet Explorer"]。

对于某一个样本，如["male","US","Internet Explorer"]，我们需要将这个分类值的

① 参考：https://blog.csdn.net/qq_33472765/article/details/86561511

特征数字化，最直接的方法，我们可以采用序列化的方式：[0,1,3]。但是这样的特征处理并不能直接放入机器学习算法中。

对于上述问题，性别的属性是二维的，同理，地区是三维的，浏览器则是四维的。这样，我们可以采用 One-Hot 编码的方式对上述的样本["male","US","Internet Explorer"]编码，male 则对应[1,0]，同理 US 对应[0,1,0]，Internet Explorer 对应[0,0,0,1]，因此完整的特征数字化的结果为：[1,0,0,1,0,0,0,0,1]。这样导致的一个结果就是数据会变得非常稀疏。

2) Hash Encoding：哈希编码[①]

哈希算法并不是一个特定的算法而是一类算法的统称。哈希算法也叫散列算法，一般来说满足这样的关系：f(data)=key，输入任意长度的 data 数据，经过哈希算法处理后输出一个定长的数据 key。同时这个过程是不可逆的，无法由 key 逆推出 data。

如果是一个 data 数据集，经过哈希算法处理后得到 key 的数据集，然后将 keys 与原始数据进行一一映射就得到了一个哈希表。一般来说哈希表 M 符合 M[key]=data 这种形式。哈希表的好处是当原始数据较大时，我们可以用哈希算法处理得到定长的哈希值 key，那么这个 key 相对原始数据要小得多。我们就可以用这个较小的数据集来做索引，达到快速查找的目的。

因此在面对高基数类别变量时，就可以用特征哈希编码的方式将原始的高维特征向量压缩成较低维特征向量，且尽量不损失原始特征的表达能力。

但是哈希算法有一个问题，就是哈希值是一个有限集合，而输入数据则可以是无穷多个。那么建立一对一关系明显是不现实的。所以“碰撞”(不同的输入数据对应了相同的哈希值)是必然会发生的，所以一个成熟的哈希算法会有较好的抗冲突性。因此在实现哈希表的结构时也要考虑到哈希冲突的问题。

哈希算法的特点为低稀疏、高压缩。

3) Embedding：嵌入式方法

Embedding 是离散数据连续化方法。Embedding 试图寻找离散值间的关系，并将其表达为连续空间上的距离。所以 Eembedding 的关键就是明确离散值间的关系。以 NLP (Natural Language Processing，自然语言处理)(是人工智能的一个子领域)为例，我们通过预测某字 c0 周围出现各字的概率，来挖掘字间的向量关系。具体的算法有：CBOW (Continuous Bag-of-Words) 与 Skip-Gram 两种模型。CBOW 的意思是，一个句子中，抠掉一个字 c0，根据上下文几个字，来预测 c0 是什么，条件概率是 P[c0|Context(c0)]。Skip-Gram 反过来，根据 c0，预测句子中其他某个字出现的概率：P[ci|c0]。这两个模型都是把问题转化为了分类问题：输入 c0 或 Context(c0)的词向量，求每个字出现的概率 P(ci)。我们可以简单理解为：如果两个不同的字输出的 P(ci)分布越接近，则这两个字就越接近，它们的词向量距离就越小[②]。

① 参考：https://blog.csdn.net/xzx1232010/article/details/83026276

② 参考：https://www.jianshu.com/p/374ee5193ff8

4）基于计数的 Encoding(编码)

基于计数的编码是将分类变量替换为训练集中的计数，其优点是对线性和非线性算法都很有价值、对异常值有一定敏感度、可以添加对数转换、适用于计数，以及可以用“1”替换看不见的变量。缺点是可能会发生冲突，如相同的编码，不同的变量。

5）特殊情况

当输入是 0、1 的二值信号，而且 0 是对所模拟的模型是有作用的，那么这时候采用 flatten 的战术，即 0 变成“0、1”，1 变成“1、0”。例如原来 64 个输入特征，flatten 后变成 128 个特征，实例参考建模攻击 PUF 的相关项目。注意 flatten 和 One-Hot 有本质区别，一个是扩展特征的长度，一个是扩展特征的维度。

2. 连续型数据

除了分类这样的离散数据，我们也会碰到诸如身高、学习成绩、资金等连续型的数据。对于连续型数据，有以下的处理方式。

(1) 缺失数据处理。在收集来的数据中，往往会出现某处数据为空或不存在的情况。一般处理方式有填 0 处理、填 NAN 处理、平均值或中位值处理等。没有特殊情况的话，一般不推荐填 0 处理，0 和空相差的意义较大，0 是有意义的。

(2) 归一化。归一化与标准化的区别为：标准化是依照特征矩阵的列来处理数据的，其通过求 z-score 的方法，将样本的特征值转换到同一量纲下；归一化是依照特征矩阵的行处理数据的，其目的在于样本向量在点乘运算或其他核函数计算相似性时，拥有统一的标准，也就是说都转换为“单位向量”。

(3) 离散化。将连续值分区，某个分区内的数据均为某个分类值。例如个人资产为连续值，处理后个人资产小于 100 万元为普通阶级，个人资产拥有 100 万～1000 万元为中产阶级，个人资产在 1000 万元以上为富人阶级等。某些情况下若取值跨度太大或者太小，可以取对数或者开方、平方等处理后再离散化。

3. 时间数据

时间数据也称时间序列或动态数据，即同一现象或数据在不同时间点或时间段的数据序列。时间数据本质上也还是一种连续型数据，但是有一些特殊的地方，例如时区、周期性。因此处理时尤其要注意特殊节假日、时区等问题。对于时间序列的处理方式一般有以下两种。

(1) 描述性时序分析：这种处理方式是通过直观的数据对比或者将数据进行可视化，根据绘制的图形进行直观观测，通过图形的方式反映出时间数据的波动特征并加以利用。

(2) 统计时序分析：通过数理统计学原理来分析时间数据。这种分析的重点在于发现时间数据值内在的相互关系，从而揭示时间数据的变化规律，进而预测时间数据的变化。比如频域分析方法就是假设任何一种时间序列都可以分解成若干不同频率的周期波动，那么就可以用不同的函数来拟合这些周期波动，从而通过这些函数的计算来预测时间数据未来的变化规律。

4. 空间数据

空间数据又称几何数据，它用来表示物体的位置、形态、大小、分布等各方面的信息，是对现世界中存在的具有定位意义的事物和现象的定量描述。根据在计算机系统中地图对现实数据的存储组织、处理方法的不同，以及空间数据本身的几何特征，空间数据又可分为图形数据和图像数据。空间数据处理方式一般有以下几种。

(1) 空间数据的坐标变化：空间数据经常需要通过投影变换，来得到经纬度参照系下的地图，对各种投影进行坐标变换的原因是输入的地图是一种投影，而输出的地图是另外一种投影。

坐标变换类型主要有以下两种。

- 几何变换：主要解决因数字化原图变形等原因引起的误差，进行几何上的调整。
- 坐标系转换：主要解决各种设备、软件中坐标不一致问题。比如各种导航地图中相同位置的坐标系是不同的。

(2) 空间数据结构的转换：一般来说，空间数据的采集使用的是矢量数据结构，而空间数据的分析则主要采用栅格数据这样有利于加快数据的分析速度。为了有效利用不同数据结构的优点，就有必要进行数据结构之间的转换。比如一条线的矢量结构由一系列的坐标对表示，转换为栅格结构时，只需要把序列中的坐标对变为栅格的行列坐标，并且根据栅格的精度，以及两点间直线方程在栅格之间插入一系列栅格点就可以了。

3.3 数据准备的主要内容

3.3.1 数据清洗

数据清洗是指发现并纠正数据文件中可识别错误的最后一道程序，包括检查数据一致性、处理无效值和缺失值等。与问卷审核不同，录入后的数据清理一般是由计算机完成而不是人工完成。

1. 处理缺失值

缺失值是指粗糙数据中由于缺少信息而造成的数据的聚类、分组、删失或截断。它指的是现有数据集中某个或某些属性的值是不完全的。

缺失值产生的原因多种多样，主要分为机械原因和人为原因。

机械原因是机械故障导致数据收集或保存的失败造成的数据缺失，如数据存储失败、存储器损坏等导致某段时间数据未能收集(对于定时数据采集而言)。

人为原因是人的主观失误、历史局限或有意隐瞒造成的数据缺失，例如，在市场调查中被访人拒绝透露相关问题的答案，或者回答的问题是无效的，或数据录入人员失误漏录了数据等。

2. 缺失值类型

缺失值从缺失的分布来讲可以分为完全随机缺失、随机缺失和完全非随机缺失。

(1) 完全随机缺失(Missing Completely At Random,MCAR):指的是数据的缺失是随机的,数据的缺失不依赖于任何不完全变量或完全变量。

(2) 随机缺失(Missing At Random,MAR):指的是数据的缺失不是完全随机的,即该类数据的缺失依赖于其他完全变量。

(3) 完全非随机缺失(Missing Not At Random,MNAR):指的是数据的缺失依赖于不完全变量自身。

从缺失值的所属属性来讲可以分为单值缺失、任意缺失和单调缺失。

(1) 单值缺失:如果所有的缺失值都是同一属性,那么这种缺失称为单值缺失。

(2) 任意缺失:如果缺失值属于不同的属性,称为任意缺失。

(3) 单调缺失:对于时间序列类的数据,可能存在随着时间变化而发生的缺失,这种缺失称为单调缺失。

3. 处理方法

数据集的缺失值是由于分析错误、缺失而没有记录的观察值。如果缺失值出现,确定的算法可能就无效了或者得不到期望的结果。缺失值比其他值更能影响模型。尽管有些模型可以处理缺失值,但是对缺失值比较敏感(某一变量的缺失可能得到不好的预测结果)。

处理缺失值的经典方法有以下几种。

1) 删除:删除含有缺失值的记录

删除主要有简单删除法和权重法。简单删除法是对缺失值进行处理的最原始方法,它将存在缺失值的个案删除。如果数据缺失问题可以通过简单的删除小部分样本来达到目标,那么这个方法是最有效的。当缺失值的类型为非完全随机缺失时,可以通过对完整的数据加权来减小偏差。把数据不完全的个案标记后,将完整的数据个案赋予不同的权重,个案的权重可以通过 logistic 或 probit 回归求得。如果解释变量中存在对权重估计起决定性因素的变量,那么这种方法可以有效减小偏差;如果解释变量和权重并不相关,则它并不能减小偏差。对于存在多个属性缺失的情况,就需要对不同属性的缺失组合标记不同的权重,这将大大增加计算的难度,降低预测的准确性,这时权重法并不理想。

2) 可能值插补缺失值

它的思想来源是以最可能的值来插补缺失值比全部删除不完全样本所产生的信息丢失要少。在数据挖掘中,面对的通常是大型的数据库,它的属性有几十个甚至几百个,因为一个属性值的缺失而放弃大量的其他属性值,这种删除是对信息的极大浪费,所以产生了以可能值对缺失值进行插补的思想与方法。常用的有如下几种方法。

(1) 虚拟替换:利用虚拟值替换缺失值。例如,不知道的类别或者数值 0。

(2) 均值替换:如果缺失值是数值型的,利用均值替换。

(3) 频数替换:如果缺失值是类别的,利用出现最多的项替换。

(4) 回归替换:利用回归方法得到回归值替换缺失值。

4. 异常值处理

异常值(Outlier)是指样本中的个别值,其数值明显偏离它(或它们)所属样本的其余观测值。一组测定值中与平均值的偏差超过两倍标准差的测定值,与平均值的偏差超过三倍标准差的测定值,称为高度异常的异常值。异常值可能通过扭曲预测模型而带来问题。

判断异常值的统计学原则如下所示。

(1) 上侧情形:异常值为高端值;

(2) 下侧情形:异常值为低端值;

(3) 双侧情形:异常值在两端可能出现极端值。

异常值处理一般分为以下几个步骤[①]:异常值检测、异常值筛选、异常值确定。其中异常值检测的方法主要有:简单统计分析(例如观察极大/小值)、箱型图和 3σ 原则。

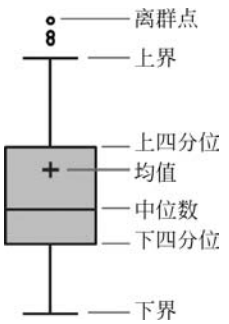

图 3-4 箱型图

(1) 简单统计分析。简单统计分析是对属性值进行一个描述性的统计,从而查看哪些值是不合理的。例如对年龄这个属性进行规约:年龄的区间在 0～200,如果样本中的年龄值不在该区间内,则表示该样本的年龄属性属于异常值。

(2) 箱型图。箱型图提供了一个识别异常值的标准,即大于或小于箱型图设定的上下界的数值即为异常值,箱型图如图 3-4 所示。上四分位设为 U,表示的是所有样本中只有 1/4 的数值大于 U;同理,下四分位设为 L,表示的是所有样本中只有 1/4 的数值小于 L。设上四分位与下四分位的插值为 IQR,即:$IQR=U-L$,上界为 $U+1.5IQR$,下界为:$L-1.5IQR$。箱型图选取异常值比较客观,在识别异常值方面有一定的优越性。

(3) 3σ 原则。

- 当数据服从正态分布时,如图 3-5 所示,根据正态分布的定义可知,距离平均值 3σ 之外的概率为 $P(|x-\mu|>3\sigma)<=0.003$,这属于极小概率事件,在默认情况下我们可以认定,距离超过平均值 3σ 的样本是不存在的。因此,当样本距离平均值大于 3σ,则认定该样本为异常值。

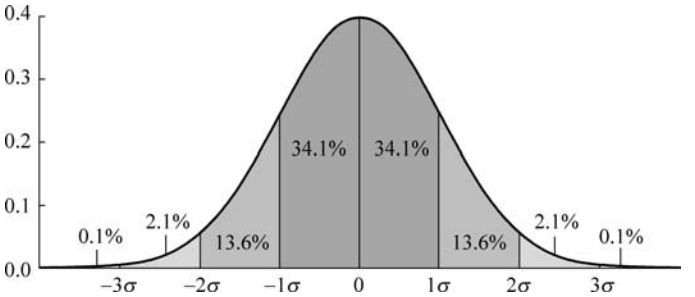

图 3-5 3σ 分布

① 参考:https://blog.csdn.net/xzfreewind/article/details/77014587

- 当数据不服从正态分布时，可以通过远离平均距离多少倍的标准差来判定，多少倍的取值需要根据经验和实际情况来决定。

异常值的处理方法常用的有如下四种。

(1) 删除含有异常值的记录。

(2) 将异常值视为缺失值，交给缺失值处理方法来处理。

(3) 用平均值来修正。

(4) 不处理。

应该强调的是，如何判定和处理异常值，需要结合实际。

5. 离群点

离群点也称为歧异值，有时也称其为野值。

概括来说，离群点是由于系统受到外部干扰而造成的。但是，形成离群点的系统外部干扰是多种多样的。首先可能是采样中的误差，如记录的偏误、工作人员出现笔误、计算错误等，都有可能产生极端大值或者极端小值。其次可能是被研究现象本身由于受到各种偶然非正常的因素影响而引起的，例如。在人口死亡序列中，由于某年发生了地震，使该年度死亡人数剧增，形成离群点；在股票价格序列中，由于受某项政策出台或某种消息的刺激，都会出现极增、极减现象，变现为序列中的离群点。

6. 离群点检测

离群点检测就是通过多种检测方法找出其行为不同于预期对象的数据点的过程。

根据正常数据和离群点的假定分类，可以分为以下 4 种方法。

1) 基于统计的方法

基于统计的离群点检测一般遵循以下思路：设定数据集的分布模型——不和谐检验——发现离群点。离群点概率定义：离群点是少数异常于正常数据集的数据对象，在概率分布模型中出现的概率较低。因此可以通过检测低概率的数据对象或数据样本，不过缺点也较为明显，低概率出现的样本不一定也是离群点(例如进货客户群中，进货量大的客户虽然少，但是也是我们需要的对象)。

2) 基于邻近性的方法

离群点，一个“离”字表现其特点，在特征空间中，离群点对象与其最近邻之间的邻近性显著偏离数据集中其他对象与它们自己的最近邻之间的邻近性。例如，使用数据对象的三个最近邻来进行建模，那么 R 区域里面的显著不同于该数据集的其他对象点；对应 R 中的对象，它们的第二个、第三个最近邻都显著比其他对象的更远(超出一定的标差)，因此可以将 R 区域中的对象作一个标记为基于邻近性的离群点。

3) 基于聚类的方法

通过考查对象与簇之间的关系检测离群点，换而言之，离群点是一个对象，它属于小的稀疏簇或者不属于任何簇。主要有如下几种考查方法。

(1) 该对象属于某个簇吗？如果不属于，则被识别为离群点(例如群居动物，山羊兔子成群居住和迁移，那么这些数据对象会划分为一个簇，这样可以把不属于这些簇的数据

对象识别为离群点)。

(2) 该对象与最近的簇之间的距离很远吗? 如果远,则被识别为离群点。

(3) 该对象是小簇或稀疏簇的一部分吗? 如果是,则该簇内所有对象被识别为离群点。

4) 基于分类的方法

如果训练数据中有类标号,则可以将其视为分类问题,处理该问题的思路一般是:训练一个可以区分"正常数据"和离群点的分类模型(一个人到银行是否办理贷款业务,"办理"与"不办理"就是两个类标号)。通常使用一类模型(One-Class Model),也就是构造一个仅仅描述正常类的分类器,这样不属于正常类的样本就是离群点,仅使用正常类检测离群点可以检测不靠近训练集中的离群点的新离群点。这样,当一个新离群点进来时,只要它位于正常类的决策边界内就为正常点,在决策边界外就为离群点。

7. 离群点的处理

具体如何处理离群点应该视情况而定。

(1) 保持离群点:一些数据中可能是真实值的离群点没有必要必须从数据中移除。在一些应用中,离群点会提供一些决定性的信息。例如,在一个信用卡欺诈检测的 App 中,离群点可以提供陷入消费者习惯购买模式之外的异常模式。

(2) 移除离群点:有两种方法移除离群点。

- 修改或者截断,其示例如图 3-6 所示。

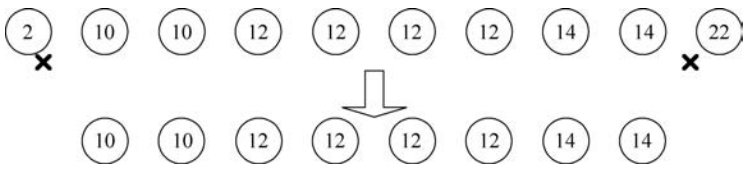

图 3-6 修改或截断示例

- 替换,其示例如图 3-7 所示。

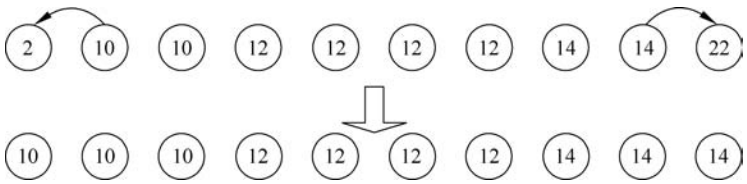

图 3-7 转换示例

修改或截断小于 5%的数据点不会对假设结果产生太大的影响。

大于 5%的数据点可能影响输出结果,主要有:削弱分析的力度、使样本缺少代表性、可能影响正常数据。考虑到数据转换,选择一个可替代的结果变量或者数据分析技术。

3.3.2　数据归一化

数据归一化(标准化)处理是数据挖掘的一项基础工作,不同评价指标往往具有不同的量纲和量纲单位,这样的情况会影响数据分析的结果,为了消除指标之间的量纲影响,需要进行数据归一化处理,以解决数据指标之间的可比性。原始数据经过数据归一化处理后,各指标处于同一数量级,适合进行综合对比。

常用的归一化方法包含如下几种。

1. Min-Max 归一化

Min-Max 归一化也称为离差标准化,是对原始数据的线性变换,使结果值映射到 0～1。转换函数如下:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

其中:$X_{\max}$ 为样本数据的最大值;$X_{\min}$ 为样本数据的最小值。这种方法的缺陷就是当有新数据加入时,可能导致 $X_{\max}$ 和 $X_{\min}$ 变化,需要重新定义。

2. Z-score 归一化(标准归一化)

根据数据的均值和方差归一化:利用数据平均值的差除以数据的方差。特征标准化使得数据具有 0 均值和标准方差。公式如下:

$$Z = \frac{X - \mu}{\sigma}$$

其中:μ 为所有样本数据的均值;σ 为所有样本数据的标准差。

3. Decimal 归一化

将数据中带有小数点的特征值去掉小数点。注意:数据集的不同度量尺度可能是有问题的,一些特定的机器学习算法是要求数据具有相同的比例的。

3.3.3　数据离散化

1. 离散化的基本概念

离散化,是把无限空间中有限的个体映射到有限的空间中去,以此来提高算法的时空效率。通俗地说,离散化是在不改变数据相对大小的条件下,对数据进行相应的缩小。示例如下。

- 原数据:1,999,100000,15;处理后:1,3,4,2。
- 原数据:{100,200},{20,50000},{1,400};处理后:{3,4},{2,6},{1,5}。

有些数据挖掘算法,特别是某些分类算法(如朴素贝叶斯),要求数据是分类属性形式(类别型属性),这样常常需要将连续属性变换成分类属性(离散化,Discretization)。另外,如果一个分类属性(或特征)具有大量不同值,或者某些出现不频繁的值,则对于某些

数据挖掘任务来说，通过合并某些值来减少类别的数目可能是有益的。

与特征选择一样，最佳的离散化方法是对于用来分析数据的数据挖掘算法，产生最好结果的方法，而直接使用这种判别标准通常是不实际的，因此，离散化一般需要满足这样一种判别标准，它与所考虑的数据挖掘任务的性能好坏直接相关。

通常离散化应用于分类或关联分析中所使用的属性上。一般来说，离散化的效果取决于所使用的算法，以及用到的其他属性。然而，属性离散化通常单独考虑。

连续属性变换为类别属性涉及两个子任务。

(1) 决定需要多少个类别值；

(2) 确定如何将连续属性映射到这些分类值。

在第一步中，将连续属性值排序后，通过指定 $n-1$ 个分割点(Split Point)把它们分成 n 个区间；在第二步中，将一个区间中的所有值映射到相同的类别上。因此，离散化问题就是决定选择多少个分割点和确定分割点位置的问题，结果可以用区间集合 $\{(x_0,x_1],(x_1,x_2],\cdots,(x_{n-1},x_n]\}$ 表示，其中 x_0 和 x_1 可以分别为 $-\infty$ 和 $+\infty$，或者用一系列不等式 $x_0<x\leqslant x_1,\cdots,x_{n-1}<x<x_n$ 表示。

2. 离散化处理的一般过程

对连续特征进行离散化处理，一般经过以下步骤。

(1) 对此特征进行排序。特别是对于大数据集，排序算法的选择要有助于节省时间、提高效率、减少离散化的整个过程的时间开支及复杂度。

(2) 选择某个点作为候选断点。用所选取的具体离散化方法的尺度衡量此候选断点是否满足要求。

(3) 选择下一个候选断点。若候选断点满足离散化的衡量尺度，则对数据集进行分裂或合并，再选择下一个候选断点，并重复步骤(2)、(3)。

(4) 停止离散化过程。当离散算法存在停止准则时，如果满足停止准则，则不再进行离散化过程，从而得到最终的离散结果。

关于离散结果的好坏，仍取决于模型的效果。

3. 离散化方法

离散化方法可以将连续数值转换为分类属性或者间隔值离散化数据图，如图 3-8 所示。

连续数据离散化的原则有基于等宽度、等频率或优化离散三种方法。

1) 等宽法

等宽法即是将属性值分为具有相同宽度的区间，区间的个数 k 根据实际情况来决定。例如属性值区间为[0,60]，最小值为 0，最大值为 60，我们要将其分为三等分，则区间被划分为[0,20] [21,40] [41,60]，每个属性值对应属于它的那个区间。

2) 等频法

等频法又称为等高法，将属性值均匀分为 n 等份，每份内包含的观察点数相同。例如有 60 个样本，我们要将其分为 $k=3$ 部分，则每部分的长度为 20 个样本。

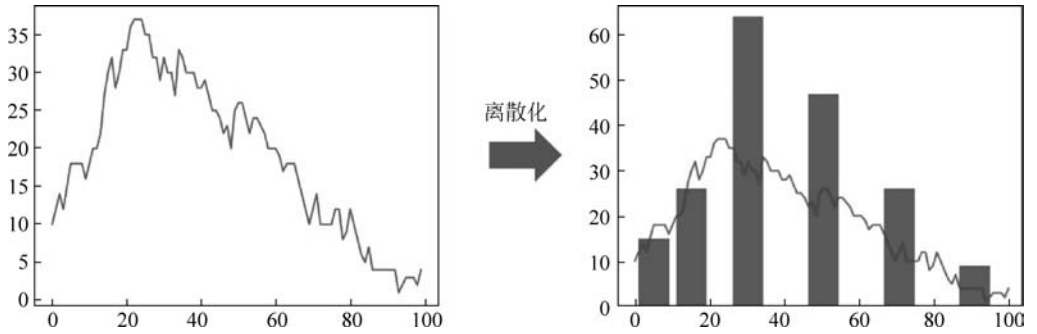

图 3-8　连续数值离散化

等宽和等频法的示意图如图 3-9 所示。

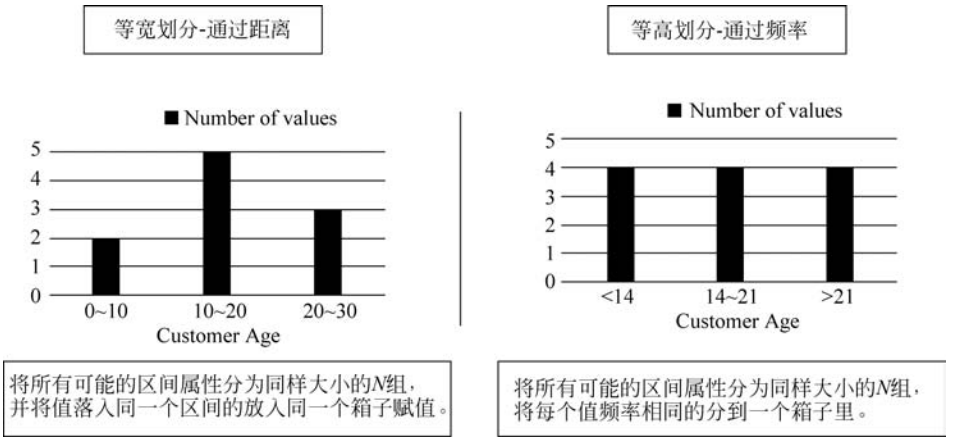

图 3-9　等宽和等高划分

3）优化离散

优化离散有卡方检验和信息增益两类方法。

（1）卡方检验方法。统计样本的实际观测值与理论推断值之间的偏离程度，卡方值越大，越不符合；卡方值越小，偏差越小，越趋于符合。

- 分裂方法：找到一个分裂点看左右两个区间在目标值上分布是否有显著差异，有显著差异就分裂，否则就忽略。这个点可以找差异最大的点。
- 合并方法：先划分小单元区间，按顺序合并在目标值上分布不显著的相邻区间，直到收敛。

（2）信息增益方法。

- 分裂方法：找到一个分裂点看左右两个区间，看分裂前后信息增益变化的阈值，如果差值超过阈值（正值，分裂前-分裂后信息熵），则分裂。每次找差值最大的点做分裂点，直到收敛。
- 合并方法：先划分区间，如果单元区间很小，则按顺序合并信息增益小于阈值的相邻区间，直到收敛。

3.3.4 数据降维

1. 基本概念

当分析的数据量非常庞大时，通常比较好用的减少数据量的方法是将数据减少维度而且使数据变得更加具有代表性和容易处理。这将促进数据的理解、探索并减少相应特征工程的工作量。大量的数据会导致算法花费更长的运行时间和更大的计算量和内存需求。在总的数据训练开始之前可以拿一些具有代表性的少量数据去训练，这样可能会更快地探索和验证方案。

在实际的机器学习项目中，特征选择/降维是必须进行的，因为在数据中存在以下几个 方面的问题。

(1) 数据的多重共线性：特征属性之间存在着相互关联关系。多重共线性会导致解的空间不稳定，从而导致模型的泛化能力弱。

(2) 高维空间样本具有稀疏性，导致模型比较难找到数据特征。

(3) 过多的变量会妨碍模型查找规律。

(4) 仅仅考虑单个变量对于目标属性的影响可能忽略变量之间的潜在关系。

通过特征选择/降维的目的是：减少特征属性的个数、确保特征属性之间是相互独立的。

如何给数据降维？有很多将数据降维的方法，以使得数据更容易处理。依赖于数据的大小和主要特征，主要有以下方法。

(1) 记录取样：只从数据样本中取出具有代表性的数据。

(2) 特征取样：只选择比较重要的几个特征。

(3) 聚合：将数据分为几组并记录每组数据的条数。例如：可以将过去 20 年连锁餐厅的每日收入聚合为月收入以减少数据量。

2. 降维方法

机器学习领域中所谓的降维就是指采用某种映射方法，将原高维空间中的数据点映射到低维度的空间中。降维的本质是学习一个映射函数 $f: x \rightarrow y$，其中 x 是原始数据点的表达，是目前最多使用的向量表达形式。y 是数据点映射后的低维向量表达，通常 y 的维度小于 x 的维度(当然提高维度也是可以的)。f 可能是显式的或隐式的、线性的或非线性的。

1) 主成分分析

主成分分析(Principal Component Analysis，PCA)是一种数学变换的方法，它把给定的一组相关变量通过线性变换转成另一组不相关的变量，这些新的变量按照方差依次递减的顺序排列。在数学变换中保持变量的总方差不变，使第一变量具有最大的方差，称为第一主成分；第二变量的方差次大，并且和第一变量不相关，称为第二主成分。以此类推，I 个变量就有 I 个主成分。

若 $\boldsymbol{L}_i$ 为 p 维正交化向量($\boldsymbol{L}_i \cdot \boldsymbol{L}_i = 1$)，$\boldsymbol{Z}_i$ 为 $\boldsymbol{L}$ 向量对应的另外一组不相关的向量，

$\boldsymbol{Z}_i$ 之间互不相关且按照方差由大到小排列，则称 $\boldsymbol{Z}_i$ 为 X 的第 I 个主成分。设 X 的协方差矩阵为 $\boldsymbol{\Sigma}$，则 $\boldsymbol{\Sigma}$ 必为半正定对称矩阵，求特征值 λ_i（按从大到小排序）及其特征向量，可以证明，λ_i 所对应的正交化特征向量，即为第 I 个主成分 $\boldsymbol{Z}_i$ 所对应的系数向量 $\boldsymbol{0}$，而 $\boldsymbol{Z}_i$ 的方差贡献率定义为 $\lambda_i/\boldsymbol{\Sigma}\lambda_j$，通常要求提取的主成分的数量 k 满足 $\boldsymbol{\Sigma}\lambda_k/\boldsymbol{\Sigma}\lambda_j>0.85$。

进行主成分分析后，还可以根据需要进一步利用 K-L 变换（霍特林变换）对原数据进行投影变换，达到降维的目的。

主成分分析算法的基本原理就是将一个矩阵中的样本数据投影到一个新的空间中去。对于一个矩阵来说，将其对角化即是产生特征根及特征向量的过程，也是将其在标准正交基上投影的过程，而特征值对应的即为该特征向量方向上的投影长度，因此该方向上携带的原有数据的信息越多。

分析步骤可以分成以下 6 个步骤。

- 将原始数据按行排列组成矩阵 $\boldsymbol{X}$；
- 对 $\boldsymbol{X}$ 进行数据标准化，使其均值变为零；
- 求 $\boldsymbol{X}$ 的协方差矩阵 $\boldsymbol{C}$；
- 将特征向量按特征值由大到小排列，取前 k 个按行组成矩阵 $\boldsymbol{P}$；
- 通过计算 $\boldsymbol{Y}=\boldsymbol{PX}$，得到降维后数据 $\boldsymbol{Y}$；
- 用下式计算每个特征根的贡献率 V_i：

$$V_i=x_i/(x_1+x_2+\cdots+x_n)$$

根据特征根及其特征向量解释主成分的物理意义。

举例来说，如果二维平面有 5 个点，则可以用 2×5 的矩阵 $\boldsymbol{X}$ 来表示：

$$\boldsymbol{X}=\begin{bmatrix}1 & 2 & 3 & 4 & 5\\ 1 & 3 & 2 & 5 & 4\end{bmatrix}$$

对 $\boldsymbol{X}$ 进行归一化，使 $\boldsymbol{X}$ 每行减去其对应的均值，得到：

$$\boldsymbol{X}=\begin{bmatrix}-2 & -1 & 0 & 1 & 2\\ -2 & 0 & -1 & 2 & 1\end{bmatrix}$$

求 $\boldsymbol{X}$ 的协方差矩阵：

$$\boldsymbol{C}=\frac{1}{5}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}=\begin{bmatrix}2 & 1.6\\ 1.6 & 2\end{bmatrix}$$

求解 $\boldsymbol{C}$ 的特征值，利用线性代数知识或是 MATLAB 中 eig 函数可以得到：

$$\lambda_1=0.4,\quad \lambda_2=3.6$$

对应的特征向量分别是：

$$\boldsymbol{\phi}_1=\begin{pmatrix}-0.7071\\ 0.7071\end{pmatrix},\quad \boldsymbol{\phi}_2=\begin{pmatrix}0.7071\\ 0.7071\end{pmatrix}$$

将原数据降为一维，选择最大的特征值对应的特征向量，因此 $\boldsymbol{P}$ 为：

$$\boldsymbol{P}=[0.7071\quad 0.7071]$$

降维后的数据：

$$\boldsymbol{Y}=\boldsymbol{PX}=[-2.8284\quad -0.7071\quad -0.7071\quad 2.1213\quad 2.1213]$$

2）线性判别分析

线性判别分析（Linear Discriminant Analysis，LDA）（也有叫作 Fisher Linear Discriminant）是一种有监督的（Supervised）线性降维算法。与主成分分析算法保持数据信息不同，线性判别分析算法是为了使得降维后的数据点尽可能地容易被区分。

线性判别分析算法的主要原理为：

- 同类的数据点尽可能地接近（Within Class）；
- 不同类的数据点尽可能地分开（Between Class）。

线性判别分析是一种经典的线性学习方法，在二分类问题上最早由 Fisher 在 1936 年提出，也称 Fisher 线性判别。

线性判别分析的思想非常朴素：给定训练样例集，设法将样例投影到一条直线上，使得同类样例的投影点尽可能接近，异样样例的投影点尽可能远离；在对新样本进行分类时，将其投影到同样的直线上，再根据投影点的位置来确定新样本的类别。

线性判别分析与方差分析（Analysis of Variance，ANOVA）和回归分析紧密相关，这两种分析方法也试图通过一些特征或测量值的线性组合来表示一个因变量。然而，方差分析使用类别自变量和连续数因变量，而判别分析连续自变量和类别因变量（即类标签）。逻辑回归和概率回归比方差分析更类似于线性判别分析，因为它们也是用连续自变量来解释类别因变量的。

线性判别分析的基本假设是自变量是正态分布的，当这一假设无法满足时，在实际应用中更倾向于使用上述的其他方法。线性判别分析也与主成分分析和因子分析紧密相关，它们都在寻找最佳解释数据的变量线性组合。线性判别分析尝试为数据类的不同建立不同的概率分布模型。另一方面，PCA 不考虑类的任何不同，因子分析是根据不同点而不是相同点来建立特征组合的。判别的分析不同因子分析还在于，它不是一个相互依存的技术：即必须区分出自变量和因变量（也称为准则变量）的不同。在每次对自变量观察测量值都是连续量的时候，线性判别分析能有效地起作用。当处理类别自变量时，与线性判别分析相对应的技术称为判别反应分析。

3）局部线性嵌入

局部线性嵌入（Locally Linear Embedding，LLE）是一种非线性降维算法，它能够使降维后的数据较好地保持原有流形结构。局部线性嵌入算法可以说是流形学习方法中最经典的工作之一。很多后续的流形学习、降维方法都与局部线性嵌入算法有密切联系。

如图 3-10 所示，使用局部线性嵌入算法将三维数据（a）映射到二维（b）之后，映射后的数据仍能保持原有的数据流形，说明局部线性嵌入算法有效地保持了数据原有的流行结构。

但是在有些情况下局部线性嵌入算法也并不适用，如果数据分布在整个封闭的球面上，局部线性嵌入算法则不能将它映射到二维空间，且不能保持原有的数据流形。那么我们在处理数据时，首先假设数据不是分布在闭合的球面或者椭球面上。

局部线性嵌入算法认为每个数据点都可以由其近邻点的线性加权组合构造得到。其算法主要分为如下三个步骤。

- 寻找每个样本点的 k 个近邻点；

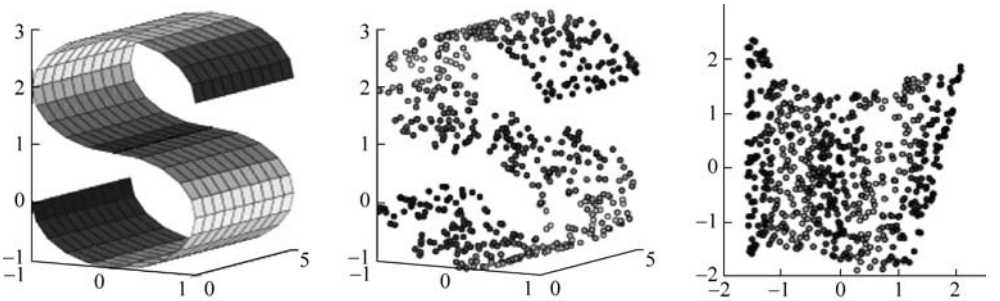

图 3-10　局部线性嵌入降维算法使用实例

- 由每个样本点的近邻点计算出该样本点的局部重建权值矩阵；
- 由该样本点的局部重建权值矩阵和其近邻点计算出该样本点的输出值。

3.3.5　文本清洗

在读取/写入文本时不适当的文本编码处理会导致信息的丢失，不经意地引入不可读的字符(如 null)也可能影响文本解析。一些非结构化的文本如推特，生产评论或者网页访问数据在被分析之前通常需要做一些准备工作。例如：

(1) 利用空格替换特殊字符和标点符号；

(2) 规范化案例；

(3) 移除重复的字符；

(4) 移除用户自定义的或者内置的停用词；

(5) 词干提取。

以英文文本为例，文本处理流程分为以下几个步骤。

1. 规范化

得到纯文本后，第一步通常就是规范化(Normalization)。在英文中，所有句子第一个单词的首字母一般是大写，有的单词也会全部字母都大写用于表示强调和区分风格，这样更易于人们理解要表达的意思，但是从计算机的角度来说是没法区别 Car、car、CAR 是否是一个意思的，因此我们一般把文本中所有字母都转换为小写或大写(通常意义上是小写)，每个词用一个唯一的字母组合来表示。小写转换和标点移除是两个最常见的文本规范化步骤，是否需要以及在哪个阶段使用这两个步骤取决于你的最终目标。

2. 分词

Token 是“符号”的高级表达，一般值是具有某种意义且无法再拆分的符号。在英文自然语言处理中，Token 通常是单独的词，因此 Tokenization(分词)就是将每个句子拆分为一系列的词。通常情况下，最简单的方法是使用 split()方法返回词列表。默认情况下是将一段话在空格字符处拆分，除了空格，也包括其他标签、新行等。同样也可以使用可选参数对其进行控制。

3. 停用词处理

停用词(Stop Words)是无含义的词,例如 is、our、the、in、at 等。它们不会给句子增加太多含义,停用词是频率非常多的一些单词。为了减少我们要处理的词汇量,从而降低后续程序的复杂度,需要清除停用词。

4. 标注词性

识别词在句子中的用途有助于我们更好理解句子内容。并且,标注词性(Part-of-Speech Tagging)还可以明确词之间的关系,并识别出交叉引用。

5. 标注实体

标注实体(Named Entity Recognition)一般是名词短语,用来指代某些特定对象、人或地点,可以使用词语标记并切分(Tokenization),并进行词性标注(PoS Tagging)。

6. 词干化和词元化

为了进一步简化文本数据,我们可以将词的不同变化和变形标准化。词干化(Stemming)提取是将词还原成词干或词根的过程。例如 branching、branched、branches 等,都可以还原成 branch。总而言之,它们都表达了分成多个路线或分支的含义。这有助于降低复杂度,并同时保留词所含的基本含义。词干化是利用非常简单的搜索和替换样式规则进行的。例如,后缀 ing 和 ed 可以丢弃; yes 可以用 y 替换等。这样可能会变成不是完整词的词干,但是只要这个词的所有形式都还原成同一个词干即可。因此它们都含有共同的根本含义。

词元化(Lemmatization)提取是将词还原成标准化形式的另一种技术。在这种情况下,转换过程实际上是利用词典,将一个词的不同变形映射到它的词根。通过这种方法,我们能将较大的词形变化,如 is/was/were 还原成词根 be。词元化需要知道每个词的词性。例如,一个文本['boring','war','started']可以使用 WordNetLemmatizer().lemmatize(w,pos='v')转换为如下文本['bore','war','start']。在这个例子中 WordNetLemmatizer()默认词性是名词。但是我们可以指定词性参数,修改这个默认设置。我们传入 v 代表动词。现在,两个动词形式 boring 和 started 都被转换了。词干化和词元化的对比如图 3-11 所示。

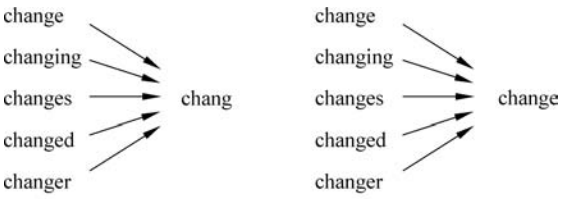

图 3-11 词干化和词元化的对比

3.4　ETL

1. ETL 的基本概念

ETL(Extraction,Transfer,Load)就是数据抽取、转换及加载,负责将分布的、异构数据源中的数据如关系数据、平面数据文件等抽取到临时中间层后进行清洗、转换、集成,最后加载到数据仓库或数据集市中,成为联机分析处理、数据挖掘的基础。为了实现这些功能,ETL 工具会进行一些功能上的扩充,例如工作流、调度引擎、规则引擎、脚本支持、统计信息等。

2. ETL 的挑战性

ETL 的功能十分具有挑战性,主要是因为源系统的性质各异。

(1) 源系统彼此悬殊,种类多样,通常需要应付多个平台上的不同操作系统。

(2) 很多源数据都是陈旧的应用系统,采用的是过时的技术。

(3) 旧系统中的数据质量各不相同,需要花费很多时间进行处理。

(4) 历史数据通常不会被保存在操作型系统中,但对于数据挖掘至关重要。

(5) 源系统之间普遍缺乏一致性。在不同的源系统中,相同的数据可能会用不同的形式来代表,且缺乏解决方法,导致不一致问题更加严重。

(6) 因为新的商业条件不断出现,源系统的结构随着时间会发生变化。ETL 功能也必须相应地调整。

(7) 大多数源系统的数据格式和类型对用户没有实际的含义,而且很多展现方式是模糊而晦涩的。

根据统计,数据挖掘工作的 50%～70%的时间花费在 ETL 上。

3. ETL 的需求和步骤

ETL 的主要步骤如图 3-12 所示。

ETL 的主要工作包括:

(1) 将几个数据源结构组合成数据仓库目标数据库中的行;

(2) 将一个源数据结构分成若干个结构放入目标数据库中的若干行;

(3) 从源系统数据字典和目录中读取数据;

(4) 从多种文件结构中读取数据,包括平面文件、索引文件、旧系统数据库;

(5) 装载大量原子事实表的细节;

(6) 为大量聚集表或事实表做聚集;

(7) 将数据从源系统平台上的一种格式转换成目标平台上的另一个格式;

(8) 将晦涩的数值改变成对用户有意义的值。

4. 数据抽取

数据抽取是从数据源中抽取数据的过程。实际应用中,数据源较多采用的是关系数

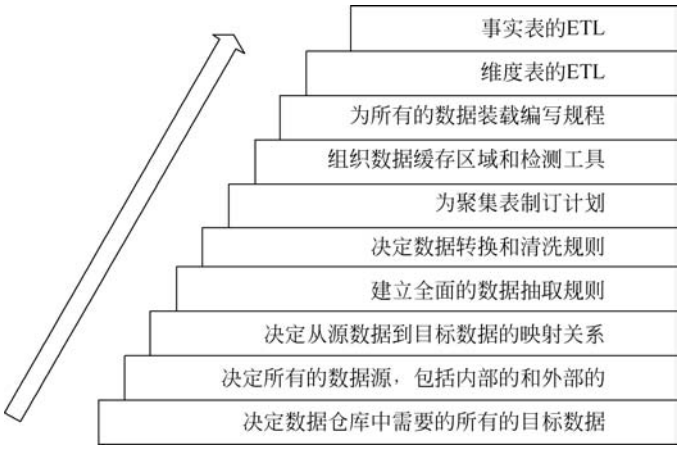

图 3-12　ETL 处理过程的主要步骤

据库。从数据库中抽取数据一般有以下两种方式。

1）全量抽取

全量抽取类似于数据迁移或数据复制，它将数据源中的表或视图的数据原封不动地从数据库中抽取出来，并转换成自己的 ETL 工具可以识别的格式。全量抽取比较简单。

2）增量抽取

增量抽取指抽取自上次抽取以来数据库中要抽取的表中新增、修改、删除的数据。在 ETL 使用过程中，增量抽取较全量抽取应用更广，如何捕获变化的数据是增量抽取的关键。对捕获方法一般有两点要求：准确性，能够将业务系统中的变化数据准确地捕获；性能，尽量减少对业务系统造成太大的压力，影响现有业务。目前增量数据抽取中常用的捕获变化数据的方法有以下几种。

（1）触发器：在要抽取的表上建立需要的触发器，一般要建立插入、修改、删除三个触发器，每当源表中的数据发生变化，就被相应的触发器将变化的数据写入一个临时表，抽取线程从临时表中抽取数据。触发器方式的优点是数据抽取的性能较高，缺点是要求在业务数据库中建立触发器，对业务系统有一定的性能影响。

（2）时间戳：它是一种基于递增数据的增量数据捕获方式，在源表上增加一个时间戳字段，系统更新修改表数据时，同时修改时间戳字段的值。当进行数据抽取时，通过比较系统时间与时间戳字段的值来决定抽取哪些数据。有的数据库的时间戳支持自动更新，即表的其他字段的数据发生改变时，自动更新时间戳字段的值。有的数据库不支持时间戳的自动更新，这就要求业务系统在更新业务数据时，手工更新时间戳字段。同触发器方式一样，时间戳方式的性能也比较好，数据抽取相对清楚简单，但对业务系统也有很大的倾入性（加入额外的时间戳字段），特别是对不支持时间戳的自动更新的数据库，还要求业务系统进行额外的更新时间戳操作。另外，无法捕获对时间戳以前数据的删除和更新操作，在数据准确性上受到了一定的限制。

（3）全表删除插入方式：每次 ETL 操作均删除目标表数据，由 ETL 全新加载数据。优点是 ETL 加载规则简单、速度快；缺点是对于维表加外键不适应，当业务系统产生删

除数据操作时，综合数据库将不会记录到所删除的历史数据，不可以实现数据的递增加载，同时对于目标表所建立的关联关系，需要重新创建。

(4) 全表比对：典型的全表比对的方式是采用 MD5 校验码。ETL 工具事先为要抽取的表建立一个结构类似的 MD5 临时表，该临时表记录源表主键以及根据所有字段的数据计算出来的 MD5 校验码。每次进行数据抽取时，对源表和 MD5 临时表进行 MD5 校验码的比对，从而决定源表中的数据是新增、修改还是删除，同时更新 MD5 校验码。MD5 方式的优点是对源系统的倾入性较小(仅需要建立一个 MD5 临时表)，但缺点也是显而易见的，与触发器和时间戳方式中的主动通知不同，MD5 方式是被动地进行全表数据的比对，性能较差。当表中没有主键或唯一列且含有重复记录时，MD5 方式的准确性较差。

(5) 日志对比：通过分析数据库自身的日志来判断变化的数据。Oracle 的改变数据捕获(Changed Data Capture，CDC)技术是这方面的代表。改变数据捕获特性是在 Oracle9i 数据库中引入的。改变数据捕获能够帮助你识别从上次抽取之后发生变化的数据。利用它在对源表进行插入、更新或删除等操作的同时就可以提取数据，并且变化的数据被保存在数据库的变化表中。这样就可以捕获发生变化的数据，然后利用数据库视图以一种可控的方式提供给目标系统。改变数据捕获的体系结构基于发布者/订阅者模型，发布者捕捉变化数据并提供给订阅者，订阅者使用从发布者那里获得的变化数据。通常，改变数据捕获系统拥有一个发布者和多个订阅者。发布者首先需要识别捕获变化数据所需的源表，然后，它捕捉变化的数据并将其保存在特别创建的变化表中，它还使订阅者能够控制对变化数据的访问。订阅者需要清楚自己感兴趣的是哪些变化数据，一个订阅者可能不会对发布者发布的所有数据都感兴趣，订阅者需要创建一个订阅者视图来访问经发布者授权可以访问的变化数据。改变数据捕获技术分为同步模式和异步模式，同步模式实时捕获变化数据并存储到变化表中，发布者与订阅都位于同一数据库中；异步模式则是基于 Oracle 的流复制技术。

ETL 处理的数据源除了关系数据库外，还可能是文件，例如 txt 文件、excel 文件、xml 文件等。一般对文件数据进行全量抽取，每次抽取前可保存文件的时间戳或计算文件的 MD5 校验码，并在下次抽取时进行比对，如果相同则可忽略本次抽取。

例如，利用订单数据提供战略信息。订单上的信息有订单数量、折扣、佣金、希望运输时间、实际运输时间、不同处理阶段时间等。涉及的维度表有产品、订单部署、运输渠道、客户。图 3-13 所示为数据源确认过程和数据源、目标之间的关系。

5. 数据转换和加工

从数据源中抽取的数据不一定完全满足目的库的要求，例如数据格式的不一致、数据输入错误、数据不完整等，因此有必要对抽取出的数据进行数据转换和加工。

数据的转换和加工可以在 ETL 引擎中进行，也可以在数据抽取过程中利用关系数据库的特性同时进行。

数据转化基本任务包括以下几个方面。

(1) 选择。从源系统得到的整个记录或部分记录。通常构成抽取功能本身的一

部分。

（2）分离或合并。包括数据处理类型。

（3）转化。多种对单独字段的基本转化：标准化和可理解化。

（4）汇总。最细事务粒度上的前期汇总。

（5）丰富。从多个源字段构成一个目标字段时，创建一个更好的数据视图。

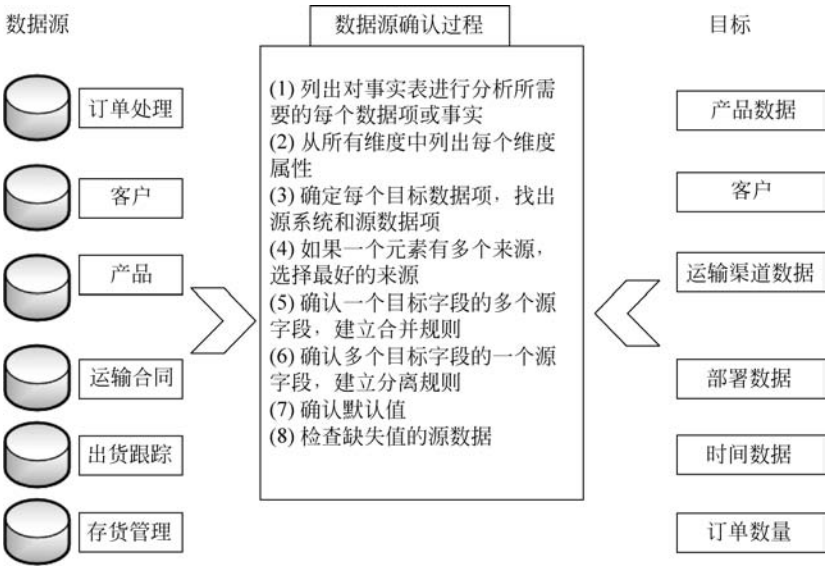

图 3-13　数据确认过程

主要转化类型包括以下几种。

（1）格式修正。包括数据类型与字段长度。

（2）字段的解码。使得晦涩的值变得易于理解和有意义。

（3）计算值和导出值。

（4）单个字段的分离。姓和名、邮编和地址。

（5）信息合并。从不同源系统中得到某个新的实体的过程。

（6）特征集合转化。编码的转化：ASCII 码、BCD 码、Unicode、Big5、GB2312 等。

（7）度量单位的转化。

（8）日期、时间格式的转化。

（9）汇总。

（10）键重构。

转化方式包括以下几种。

（1）ETL 引擎中的数据转换和加工。ETL 引擎中一般以组件化的方式实现数据转换。常用的数据转换组件有字段映射、数据过滤、数据清洗、数据替换、数据计算、数据验证、数据加解密、数据合并、数据拆分等。这些组件如同一条流水线上的一道道工序，它们是可插拔的，且可以任意组装，各组件之间通过数据总线共享数据。同时 ETL 工具还提供了脚本支持，使得用户可以以一种编程的方式定制数据的转换和加工行为。

相比在数据库中加工，性能较高，但不容易进行修改和清晰辨认。

(2) 在数据库中进行数据加工。关系数据库本身已经提供了强大的 SQL、函数来支持数据的加工，如在 SQL 查询语句中添加 where 条件进行过滤，可以在查询中重命名字段名与目的表进行映射，提供多种函数进行复杂运算等。

相比在 ETL 引擎中进行数据转换和加工，直接在 SQL 语句中进行转换和加工更加简单清晰，但依赖 SQL 语句，有些数据加工通过 SQL 语句可能无法实现，对于 SQL 语句无法处理的可以交由 ETL 引擎处理。

6. 数据装载

将转换和加工后的数据装载到目的库中通常是 ETL 过程的最后步骤。装载数据的最佳方法取决于所执行操作的类型以及需要装入多少数据。

数据装载易出现问题，主要因为：需要大量的时间，而且时间不好估计；装载的过程可能是不顺利的；实际装载数据与计划制订可能不匹配(维度表与事实表的不匹配)；不知数据准备区和数据仓库数据库分别处在何处；装载牵涉到维度表、事实表；装载需要专门的程序。

当目的库是关系数据库时，一般有以下两种装载方式。

(1) 直接 SQL 语句进行插入、修改、删除操作。

(2) 采用批量装载方法，如 sqlldr 等。

大多数情况下使用第一种方法，因为它们进行了日志记录并且是可恢复的。但是，批量装载操作易于使用，并且在装入大量数据时效率较高。使用哪种数据装载方法取决于业务系统的需要。

3.5　习题

1. 简述数据准备的主要内容。

2. 将表 3-3 中的数据规范化到区间[0,1]。

表 3-3　第 2 题数据

身高/m	体重/kg	身高/m	体重/kg
1.62	55	1.68	62
1.65	57	1.75	60
1.60	45	1.80	90
1.72	65	1.76	70
1.73	70	1.82	75

3. 用分箱法对表 3-3 中的数据进行离散化处理。

4. 已知有矩阵：$\boldsymbol{X}=\begin{bmatrix}1 & 2 & 3 & 7 & 8\\ 1 & 3 & 2 & 8 & 6\end{bmatrix}$，计算该矩阵降为一维后的数据。

5. 简述 ETL 处理的基本步骤。